UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/087,234	03/01/2002	Ajay Kumar	5681-11700	6950

7590 12/06/2006

Robert C Kowert
Conley Rose & Tayon P C
P O Box 398
Austin, TX 78767-0398

EXAMINER	
HWANG, JOON H	

ART UNIT	PAPER NUMBER
2166	

DATE MAILED: 12/06/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/087,234	KUMAR ET AL.	
	Examiner	Art Unit	
	Joon H. Hwang	2166	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *14 September 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-11,13-26,28-35 and 37* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-11,13-26,28-35 and 37* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Applicant's request for reconsideration of the finality of the rejection of the last Office action is persuasive and, therefore, the finality of that action is withdrawn.

The pending claims are 1-11, 13-26, 28-35, and 37.

Response to Arguments

2. Applicant's arguments filed in the amendment received on 1/23/06 have been fully considered but they are not persuasive.

A. The applicants argue that Montero and Goldick does not teach or suggest a distributed store configured to provide locked access to the primary state of session data to a process executing within one of a plurality of application servers and the distributed store is configured to send a lock token to the process.

The examiner respectfully traverses. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in

the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, Montero discloses a common session database (a distributed store) configured to provide access to the primary state of session data to a process executing within one of a plurality of application servers (fig. 1, section 35 on page 3, and section 40 on page 4). Montero teaches a distributed environment since the common session database is shared by the plurality of application servers (fig. 1). Montero does not explicitly disclose a locking management of the database. However, Goldick discloses a resource management in a distributed environment, wherein a plurality of clients shares the same resource (section 2 on page 1, fig. 1, and fig. 3). Goldick recognizes a problem in the distributed environment when two or more users modify the same resource simultaneously such that editions are inadvertently lost (section 5 on page 1). Thus, Goldick discloses a locking management on a resource of a server system in a shared resource distributed computing environment (fig. 1, fig. 3, section 5 on page 1, and sections 24-25 on page 3) in order to prevent the "lost update" problem. Goldick discloses the server system comprising a resource configured for access by the plurality of client nodes, wherein the server system is configured to provide locked access to the resource to one of the plurality of the client nodes, wherein, while the resource is locked for the node, other nodes cannot access the resource (sections 24-25 on page 3). Goldick discloses wherein in providing locked access to the resource to the one of the plurality of the client nodes, the server system is configured to send a lock token to the node, wherein only the node that have received a lock token can access the resource

(sections 44-45 on page 5 and fig. 3) in order to prevent data inconsistency (section 5 on page 1). Therefore, based on Montero in view of Goldick, it would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize teaching of Goldick to the system of Montero in order to prevent data inconsistency.

"Test of obviousness is not whether features of secondary reference may be bodily incorporated into primary reference's structure, nor whether claimed invention is expressly suggested in any one or all of references; rather, **test is what combined teachings of references would have suggested to those of ordinary skill in art**." In re Keller, Terry, and Davies, 208 USPQ 871 (CCPA 1981).

"Reason, suggestion, or **motivation to combine two or more prior art references** in single invention **may come from references themselves**, from **knowledge of those skilled in art** that certain references or disclosures in references are known to be of interest in particular field, or from **nature of problem to be solved**;" Pro-Mold and Tool Co. v. Great Lakes Plastics Inc. U.S. Court of Appeals Federal Circuit 37 USPQ2d 1626 Decided February 7, 1996 Nos. 95-1171, -1181.

In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., **a distributed store** configured to provide locked access to the primary state of session data to a process executing within one of a plurality of application servers and **the distributed store** configured to send a lock token to the process) are not recited in

the rejected claims 17, 20, and 31. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

B. The applicants argue that Montero, Bennet, Bender, and Eshel do not teach or suggest a distributed store configured to request the process to release the locked access.

The examiner respectfully traverses. In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., **a distributed store** configured to request the process to release the locked access) are not recited in the rejected claims 26-30, 35, and 37. Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Claim Objections

3. Claims 1, 10, 17 and 35 are objected to under 37 CFR 1.75(c), as being of improper dependent form for failing to further limit the subject matter of a previous claim. Applicant is required to cancel the claim(s), or amend the claim(s) to place the claim(s) in proper dependent form, or rewrite the claim(s) in independent form.

- "the state of a client session" in 4th line of claim 1 should be "a state of a client session";

- "the state of a client session" in 4th line of claim 10 should be "a state of a client session";
- "the state of a client session" in 4th line of claim 17 should be "a state of a client session"; and
- "the lock" in 6th line of claim 35 should be "a lock".

Double Patenting

4. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

5. A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory

double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

6. Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

7. Claims 1 and 17 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1 and 10 of U.S. Patent No. 7,093,230 in view of Hopmann et al. (U.S. Patent No. 6,499,031). Although the conflicting claims are not identical, they are not patentably distinct from each other because they are substantially similar in scope and they use the same limitations, using varying terminology. See further explanation below. Differences are bolded and omissions are underlined in following comparison tables.

Instant Application	'230 Patent
1. A system, comprising : a plurality of application servers, wherein each of the plurality of application servers is configured to access session data, wherein the session data represents the state of a client session for a client; and	1. A distributed data system, comprising: a plurality of network nodes each configured to execute one or more processes;
a distributed store comprising a primary state of the session data configured for access by the plurality of application servers, wherein the distributed store is configured to provide locked access to the primary state to a process executing within one of the plurality of application servers, wherein, while the primary state is locked for the process, other processes cannot access the primary state; wherein in providing locked access to the	a data store configured to store primary data accessible by the processes; and a lock mechanism coupled to the data store and configured to lock access to portions of the primary data, wherein the lock mechanism is configured to **grant a lock** to a requester for one of the processes for a primary data portion stored by the data store, wherein the lock mechanism is configured to prevent other processes from accessing the primary data portion while the

primary state to a process executing within one of the plurality of application servers, the distributed store is configured to **send a lock token** to the process, wherein only processes that have received a lock token can access the primary state.	requester is granted the lock; wherein each of the processes is configured to maintain a thread pool of lock management threads, wherein each of the lock management threads is configured to be a requester on behalf of its process for a lock from the lock mechanism for a portion of the primary data.
	10. The distributed data system as recited in claim 1, wherein a portion of the primary data, stored by the data store represents state information of a client-server session.

The bolded difference "sending a lock token" is taught by Hopmann in order to prevent a resource from being corrupted (i.e., a lock token is given to a requesting client and the client at a later time present the lock token to a server in order to access a locked resource, lines 35-52 in col. 2, lines 66-67 in col. 6, lines 3-21 and 34-51 in col. 8, and lines 5-6 in col. 9).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teaching of Hopmann to the system of '230 Patent in order to prevent a resource from being corrupted.

Instant Application	'230 Patent
17. A system, comprising : a plurality of application servers, wherein each of the plurality of application servers is configured to access session data, wherein the session data represents the state of a client session for a client; and a distributed store comprising a primary state of the session data configured for access by the	1. A distributed data system, comprising: a plurality of network nodes each configured to execute one or more processes; a data store configured to store primary data accessible by the processes; and

plurality of application servers; and means for providing locked access to the primary state to a process executing within one of the plurality of application servers, wherein, while the primary state is locked for the process, other processes cannot access the primary state; wherein said means for providing locked access comprises means for **sending** the process **a lock token**, wherein only processes that have received a lock token can access the primary state.	a lock mechanism coupled to the data store and configured to lock access to portions of the primary data, wherein the lock mechanism is configured to **grant a lock** to a requester for one of the processes for a primary data portion stored by the data store, wherein the lock mechanism is configured to prevent other processes from accessing the primary data portion while the requester is granted the lock; wherein each of the processes is configured to maintain a thread pool of lock management threads, wherein each of the lock management threads is configured to be a requester on behalf of its process for a lock from the lock mechanism for a portion of the primary data.
	10. The distributed data system as recited in claim 1, wherein a portion of the primary data, stored by the data store represents state information of a client-server session.

The bolded difference "sending a lock token" is taught by Hopmann in order to prevent a resource from being corrupted (i.e., a lock token is given to a requesting client and the client at a later time present the lock token to a server in order to access a locked resource, lines 35-52 in col. 2, lines 66-67 in col. 6, lines 3-21 and 34-51 in col. 8, and lines 5-6 in col. 9).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teaching of Hopmann to the system of '230 Patent in order to prevent a resource from being corrupted.

8. Claim 10 is rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1 and 6 of U.S. Patent No. 7,093,230 in

view of Jain et al. (U.S. Patent No. 6,484,185). Although the conflicting claims are not identical, they are not patentably distinct from each other because they are substantially similar in scope and they use the same limitations, using varying terminology. See further explanation below. Differences are bolded and omissions are underlined in following comparison tables.

Instant Application	'230 Patent
10. A system, comprising : a plurality of application servers, wherein each of the plurality of application servers is configured to access session data, wherein the session data represents the state of a client session for a client; and a distributed store comprising **a primary state of the session data** configured for access by the plurality of application servers, wherein the distributed store is configured to provide locked access to the primary state to a process executing within one of the plurality of application servers, wherein, while the primary state is locked for the process, **other processes cannot access the primary state**; wherein the process is configured to provide locked access to portions of the primary state to one or more threads executing within the process, wherein while a portion of the primary state is locked for one of the threads, **other threads executing within the process cannot access the particular portion of the primary state**; wherein the distributed store is configured to request the process to release the locked access, wherein the process is configured to release the locked access in response to said request.	1. A distributed data system, comprising: a plurality of network nodes each configured to execute one or more processes; a data store configured to store **primary data** accessible by the processes; and a lock mechanism coupled to the data store and configured to lock access to portions of the primary data, wherein the lock mechanism is configured to grant a lock to a requester for one of the processes for a primary data portion stored by the data store, wherein the lock mechanism is configured to prevent other processes from accessing the primary data portion while the requester is granted the lock; wherein each of the processes is configured to maintain a thread pool of lock management threads, wherein each of the lock management threads is configured to be a requester on behalf of its process for a lock from the lock mechanism for a portion of the primary data.
	6. The distributed data system as recited in claim 1, wherein each of the lock management threads is configured to hold the lock it acquires for its process until that process receives a request to release the lock.

The bolded difference "other threads executing within the process cannot access the particular portion of the primary state" is taught by Jain in order to provide atomic operations in multiprocessing system (i.e., one of threads within a process has an exclusive lock, no other thread may read from or write to shared data, lines 4-12 in col. 2, lines 35-39 in col. 2, and lines 52-57 in col. 11).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teaching of Jain to the system of '230 Patent in order to prevent a resource from being corrupted in multiprocessing system.

9. Claims 20-21, 26, 31, and 35 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 22, 26, 28, and 32 of U.S. Patent No. 7,093,230 in view of Hopmann et al. (U.S. Patent No. 6,499,031) and Jain et al. (U.S. Patent No. 6,484,185). Although the conflicting claims are not identical, they are not patentably distinct from each other because they are substantially similar in scope and they use the same limitations, using varying terminology. See further explanation below. Differences are bolded and omissions are underlined in following comparison tables.

Instant Application	'230 Patent
20. A method, comprising : a process executing within one of a plurality of application servers requesting locked access to a	22. A computer implemented method comprising: a requesting thread of a process requesting access within the process to data corresponding to

primary state of session data stored by a distributed store; and granting a lock to the process requesting locked access, wherein, while the process holds the lock, **other processes cannot access the primary state in the distributed store**; wherein said granting comprises **sending a lock token** to the process, wherein only processes that have received a lock token can access the primary state.	a portion of primary data stored in a distributed data store; in response to the thread requesting access, the process selecting a lock management thread from a lock management thread pool to request a lock for the portion of the primary data; the selected lock management thread acquiring the lock for the portion of primary data; and in response to the selected lock management thread acquiring the lock, the requesting thread accessing the portion of primary data.
21. The method as recited in claim 20, further comprising the process holding the lock granting a thread-level lock to a portion of the primary state to a thread running within the process, wherein, while the thread holds the thread-level lock, **other threads cannot access the portion of the primary state**.	

The bolded difference "sending a lock token" is taught by Hopmann in order to prevent a resource from being corrupted (i.e., a lock token is given to a requesting client and the client at a later time present the lock token to a server in order to access a locked resource, lines 35-52 in col. 2, lines 66-67 in col. 6, lines 3-21 and 34-51 in col. 8, and lines 5-6 in col. 9) and "other processes/threads cannot access the portion of the primary state" is taught by Jain in order to provide atomic operations in multiprocessing system (i.e., one of threads within a process has an exclusive lock, no other thread may read from or write to shared data, lines 4-12 in col. 2, lines 35-39 in col. 2, and lines 52-57 in col. 11).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teaching of Hopmann and Jain to the system of '230 Patent in order to prevent a resource from being corrupted in multiprocessing system.

Instant Application	'230 Patent
26. A method, comprising : a process executing within one of a plurality of application servers requesting locked access to a primary state of session data stored by a distributed store; and granting a lock to the process requesting locked access, wherein, while the process holds the lock, **other processes cannot access the primary state in the distributed store**; the process holding the lock granting a thread-level lock to a portion of the primary state to a thread running within the process, wherein, while the thread holds the thread-level lock, **other threads cannot access the portion of the primary state;** requesting the process release the locked access; and the process releasing the locked access in response to said request.	22. A computer implemented method comprising: a requesting thread of a process requesting access within the process to data corresponding to a portion of primary data stored in a distributed data store; in response to the thread requesting access, the process selecting a lock management thread from a lock management thread pool to request a lock for the portion of the primary data; the selected lock management thread **acquiring the lock** for the portion of primary data; and in response to the selected lock management thread acquiring the lock, the requesting thread accessing the portion of primary data.
	26. The method as recited in claim 22, further comprising the process holding the lock until requested to release the lock.

The bolded difference "sending a lock token" is taught by Hopmann in order to prevent a resource from being corrupted (i.e., a lock token is given to a requesting client and the client at a later time present the lock token to a server in order to access a locked resource, lines 35-52 in col. 2, lines 66-67 in col. 6, lines 3-21 and 34-51 in col. 8, and lines 5-6 in col. 9) and "other processes/threads cannot access the portion of the primary state" is taught by Jain in order to provide atomic operations in multiprocessing system (i.e., one of threads within a process has an exclusive lock, no other thread may read from or write to shared data, lines 4-12 in col. 2, lines 35-39 in col. 2, and lines 52-57 in col. 11).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teaching of Hopmann and Jain to the system of '230 Patent in order to prevent a resource from being corrupted in multiprocessing system.

Instant Application	'230 Patent
31. A computer-accessible storage medium, comprising software instructions executable to implement: receiving a request from a process executing within one of a plurality of application servers for locked access to a primary state of session data comprising one or more attributes on a distributed store; and granting locked access to the primary state to the process, wherein, while the primary state is locked for the process, **other processes cannot access the primary state**; wherein said granting locked access to the primary state comprises **sending a lock token** to the process, wherein only processes that have received a lock token can access the primary state.	28. An article of manufacture comprising program instructions executable to implement: a requesting thread of a process requesting access within the process to data corresponding to a portion of primary data stored in a distributed data store; in response to the thread requesting access, the process selecting a lock management thread from a lock management thread pool to request a lock for the portion of the primary data; the selected lock management thread **acquiring the lock** for the portion of primary data; and in response to the selected lock management thread acquiring the lock, the requesting thread accessing the portion of primary data.

The bolded difference "sending a lock token" is taught by Hopmann in order to prevent a resource from being corrupted (i.e., a lock token is given to a requesting client and the client at a later time present the lock token to a server in order to access a locked resource, lines 35-52 in col. 2, lines 66-67 in col. 6, lines 3-21 and 34-51 in col. 8, and lines 5-6 in col. 9) and "other processes/threads cannot access the portion of the primary state" is taught by Jain in order to provide atomic operations in multiprocessing system (i.e., one of threads within a process has an exclusive lock, no other thread may read from or write to shared data, lines 4-12 in col. 2, lines 35-39 in col. 2, and lines 52-57 in col. 11).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teaching of Hopmann and Jain to the system of '230 Patent in order to prevent a resource from being corrupted in multiprocessing system.

Instant Application	'230 Patent
35. A computer-accessible storage medium, comprising software instructions executable to implement: a process executing within one of a plurality of application servers requesting locked access to a primary state of session data stored by a distributed store; and the process receiving locked access, wherein, while the process holds the lock, **other processes cannot access the primary state in the distributed store**; the process granting a thread-level lock to a portion of the primary state to a thread running within the process, wherein, while the thread holds the thread-level lock, **other threads cannot access the portion of the primary state;** the process receiving a request to release the locked access; and the process releasing the locked access in response to said request.	28. An article of manufacture comprising program instructions executable to implement: a requesting thread of a process requesting access within the process to data corresponding to a portion of primary data stored in a distributed data store; in response to the thread requesting access, the process selecting a lock management thread from a lock management thread pool to request a lock for the portion of the primary data; the selected lock management thread **acquiring the lock** for the portion of primary data; and in response to the selected lock management thread acquiring the lock, the requesting thread accessing the portion of primary data.
	32. The article of manufacture as recited in claim 28, wherein the program instructions are further executable to implement the process holding the lock until requested to release the lock.

The bolded difference "sending a lock token" is taught by Hopmann in order to prevent a resource from being corrupted (i.e., a lock token is given to a requesting client and the client at a later time present the lock token to a server in order to access a locked resource, lines 35-52 in col. 2, lines 66-67 in col. 6, lines 3-21 and 34-51 in col. 8, and lines 5-6 in col. 9) and "other processes/threads cannot access the portion of the

primary state" is taught by Jain in order to provide atomic operations in multiprocessing system (i.e., one of threads within a process has an exclusive lock, no other thread may read from or write to shared data, lines 4-12 in col. 2, lines 35-39 in col. 2, and lines 52-57 in col. 11).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teaching of Hopmann and Jain to the system of '230 Patent in order to prevent a resource from being corrupted in multiprocessing system.

10. Claims 1, 10, and 28 of U.S. Patent No. 7,093,230 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1, 17, and 31 of Instant Application in view of Jain et al. (U.S. Patent No. 6,484,185). Although the conflicting claims are not identical, they are not patentably distinct from each other because they are substantially similar in scope and they use the same limitations, using varying terminology. See further explanation below. Differences are bolded and omissions are underlined in following comparison tables.

'230 Patent	Instant Application
1. A distributed data system, comprising: a plurality of network nodes each configured to execute one or more processes;	1. A system, comprising : a plurality of application servers, wherein each of the plurality of application servers is configured to access session data, wherein the session data represents the state of a client session for a client; and
a data store configured to store primary data accessible by the processes; and a lock mechanism coupled to the data store and	a distributed store comprising a primary state of the session data configured for access by the plurality of application servers, wherein the

configured to lock access to portions of the primary data, wherein the lock mechanism is configured to **grant a lock** to a requester for one of the processes for a primary data portion stored by the data store, wherein the lock mechanism is configured to prevent other processes from accessing the primary data portion while the requester is granted the lock; **wherein each of the processes is configured to maintain a thread pool of lock management threads, wherein each of the lock management threads is configured to be a requester on behalf of its process for a lock from the lock mechanism for a portion of the primary data.**	distributed store is configured to provide locked access to the primary state to a process executing within one of the plurality of application servers, wherein, while the primary state is locked for the process, other processes cannot access the primary state; wherein in providing locked access to the primary state to a process executing within one of the plurality of application servers, the distributed store is configured to **send a lock token** to the process, wherein only processes that have received a lock token can access the primary state.
10. The distributed data system as recited in claim 1, wherein a portion of the primary data, stored by the data store represents state information of a client-server session.	

The bolded difference "wherein each of the processes is configured to maintain a thread pool of lock management threads, wherein each of the lock management threads is configured to be a requester on behalf of its process for a lock from the lock mechanism for a portion of the primary data" is taught by Jain in order to provide atomic operations in multiprocessing system (i.e., one of threads within a process has an exclusive lock, no other thread may read from or write to shared data, lines 4-12 in col. 2, lines 35-39 in col. 2, and lines 52-57 in col. 11).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teaching of Jain to the system of Instant Application in order to provide atomic operations in multiprocessing system.

'230 Patent	Instant Application
1. A distributed data system, comprising: a plurality of network nodes each configured to execute one or more processes; a data store configured to store primary data accessible by the processes; and a lock mechanism coupled to the data store and configured to lock access to portions of the primary data, wherein the lock mechanism is configured to **grant a lock** to a requester for one of the processes for a primary data portion stored by the data store, wherein the lock mechanism is configured to prevent other processes from accessing the primary data portion while the requester is granted the lock; **wherein each of the processes is configured to maintain a thread pool of lock management threads, wherein each of the lock management threads is configured to be a requester on behalf of its process for a lock from the lock mechanism for a portion of the primary data.**	17. A system, comprising : a plurality of application servers, wherein each of the plurality of application servers is configured to access session data, wherein the session data represents the state of a client session for a client; and a distributed store comprising a primary state of the session data configured for access by the plurality of application servers; and means for providing locked access to the primary state to a process executing within one of the plurality of application servers, wherein, while the primary state is locked for the process, other processes cannot access the primary state; wherein said means for providing locked access comprises means for **sending** the process **a lock token**, wherein only processes that have received a lock token can access the primary state.
10. The distributed data system as recited in claim 1, wherein a portion of the primary data, stored by the data store represents state information of a client-server session.	

The bolded difference "wherein each of the processes is configured to maintain a thread pool of lock management threads, wherein each of the lock management threads is configured to be a requester on behalf of its process for a lock from the lock mechanism for a portion of the primary data" is taught by Jain in order to provide atomic operations in multiprocessing system (i.e., one of threads within a process has an exclusive lock, no other thread may read from or write to shared data, lines 4-12 in col. 2, lines 35-39 in col. 2, and lines 52-57 in col. 11).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teaching of Jain to the system of Instant Application in order to provide atomic operations in multiprocessing system.

'230 Patent	Instant Application
28. An article of manufacture comprising program instructions executable to implement: **a requesting thread of a process** requesting access within the process to data corresponding to a portion of primary data stored in a distributed data store; in response to the **thread** requesting access, the process selecting a lock management **thread** from a lock management thread pool to request a lock for the portion of the primary data; the selected lock management **thread acquiring the lock** for the portion of primary data; and in response to the selected lock management **thread** acquiring the lock, the requesting **thread** accessing the portion of primary data.	31. A computer-accessible storage medium, comprising software instructions executable to implement: receiving **a request from a process** executing within one of a plurality of application servers for locked access to a primary state of session data comprising one or more attributes on a distributed store; and granting locked access to the primary state to the **process,** wherein, while the primary state is locked for the **process**, other processes cannot access the primary state; wherein said granting locked access to the primary state comprises **sending a lock token** to the **process**, wherein only processes that have received a lock token can access the primary state.

The bolded difference "thread acquiring the lock" is taught by Jain in order to provide atomic operations in multiprocessing system (i.e., one of threads within a process has an exclusive lock, no other thread may read from or write to shared data, lines 4-12 in col. 2, lines 35-39 in col. 2, and lines 52-57 in col. 11).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teaching of Jain to the system of Instant Application in order to provide atomic operations in multiprocessing system.

11. Claims 1, 6, 22, 26, 28, and 32 of U.S. Patent No. 7,093,230 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 10, 20-21, 26, and 35 of the instant application. Although the conflicting claims are not identical, they are not patentably distinct from each other because of following reasons:

Claim 10 of the instant application contain(s) every element of claims 1 and 6 of U.S. Patent No. 7,093,230 and thus anticipate the claim(s) of the instant application. Claims of the instant application therefore are not patently distinct from the earlier patent claims and as such are unpatentable over obvious-type double patenting. A later patent/application claim is not patentably distinct from an earlier claim if the later claim is anticipated by the earlier claim.

Claims 20-21 of the instant application contain(s) every element of claim 22 of U.S. Patent No. 7,093,230 and thus anticipate the claim(s) of the instant application. Claims of the instant application therefore are not patently distinct from the earlier patent claims and as such are unpatentable over obvious-type double patenting. A later patent/application claim is not patentably distinct from an earlier claim if the later claim is anticipated by the earlier claim.

Claim 26 of the instant application contain(s) every element of claims 22 and 26 of U.S. Patent No. 7,093,230 and thus anticipate the claim(s) of the instant application. Claims of the instant application therefore are not patently distinct from the earlier patent claims and as such are unpatentable over obvious-type double patenting. A later patent/application claim is not patentably distinct from an earlier claim if the later claim

is anticipated by the earlier claim.

Claim 35 of the instant application contain(s) every element of claims 28 and 32 of U.S. Patent No. 7,093,230 and thus anticipate the claim(s) of the instant application. Claims of the instant application therefore are not patently distinct from the earlier patent claims and as such are unpatentable over obvious-type double patenting. A later patent/application claim is not patentably distinct from an earlier claim if the later claim is anticipated by the earlier claim.

"A later patent claim is not patentably distinct from an earlier patent claim if the later claim is obvious over, or anticipated by, the earlier claim. In re Longi, 759 F.2d at 896, 225 USPQ at 651 (affirming a holding of obviousness-type double patenting because the claims at issue were obvious over claims in four prior art patents); In re Berg, 140 F.3d at 1437, 46 USPQ2d at 1233 (Fed. Cir. 1998) (affirming a holding of obviousness-type double patenting where a patent application claim to a genus is anticipated by a 35 patent claim to a species within that genus). " ELI LILLY AND COMPANY v BARR LABORATORIES, INC., United States Court of Appeals for the Federal Circuit, ON PETITION FOR REHEARING EN BANC (DECIDED: May 30, 2001).

"Claim 12 and Claim 13 are generic to the species of invention covered by claim 3 of the patent. Thus, the generic invention is "anticipated" by the species of the patented invention. Cf., Titanium Metals Corp. v. Banner, 778 F.2d 775, 227 USPQ 773 (Fed. Cir. 1985) (holding that an earlier species disclosure in the prior art defeats any generic claim) 4. This court's predecessor has held that, without a terminal disclaimer,

the species claims preclude issuance of the generic application. In re Van Ornum, 686 F.2d 937, 944, 214 USPQ 761, 767 (CCPA 1982); Schneller, 397 F.2d at 354. Accordingly, absent a terminal disclaimer, claims 12 and 13 were properly rejected under the doctrine of obviousness-type double patenting." (In re Goodman (CA FC) 29 USPQ2d 2010 (12/3/1993).

Claim Rejections - 35 USC § 112

12. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

13. Claims 6 and 13 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

The limitation "that process" in 3rd line of claim 6 is unclear and indefinite.

The limitation "that process" in 3rd line of claim 13 is unclear and indefinite.

Claim Rejections - 35 USC § 102

14. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

15. Claims 1-4, 6-9, 17, 19-20, 22-25, and 31-34 are rejected under 35 U.S.C. 102(e) as being anticipated by Dinker et al. (U.S. Patent No. 7,130,905).

The applied reference has a common assignee with the instant application. Based upon the earlier effective U.S. filing date of the reference, it constitutes prior art under 35 U.S.C. 102(e). This rejection under 35 U.S.C. 102(e) might be overcome either by a showing under 37 CFR 1.132 that any invention disclosed but not claimed in the reference was derived from the inventor of this application and is thus not the invention "by another," or by an appropriate showing under 37 CFR 1.131.

With respect to claim 1, Dinker teaches a plurality of application servers (i.e., application servers 108 in fig. 1), wherein each of the plurality of application servers is configured to access session data (i.e., an application server obtaining HTTP session data from a centralized location, lines 24-37 in col. 9), wherein the session data represents the state of a client session for a client (i.e., HTTP session data for an end user, lines 24-37 in col. 9 and lines 24-42 in col. 6). Dinker teaches a distributed store (i.e., a primary application server acting as a centralized location for storing shared data, such as HTTP session data, needed by other application servers, lines 24-37 in col. 9) comprising a primary state of the session data configured for access by the plurality of application servers, wherein the distributed store is configured to provide locked access to the primary state (i.e., the DTM service executing on the primary application server coordinates access to HTTP session data, lines 38-48 in col. 9) to a process executing within one of the plurality of application servers (i.e., a process of an application server, lines 10-43 in col. 6), wherein, while the primary state is locked for

the process, other processes cannot access the primary state (i.e., one process has write access to data then no other processes have read or write access to the data, line 58 in col. 2 thru line 9 in col. 3). Dinker teaches in providing locked access to the primary state to a process executing within one of the plurality of application servers, the distributed store is configured to send a lock token to the process (i.e., the DTM send a data structure indicating that a client has acquired the requested access rights to the client, lines 37-52 in col. 11), wherein only processes that have received a lock token can access the primary state (lines 37-52 in col. 11).

With respect to claim 2, Dinker teaches after the process has completed a current access of the primary state, the process is configured to hold locked access until after receiving a request to release the locked access (i.e., the client may still hold access right to the data by default until the DTM service reclaims the access right, lines 9-19 in col. 12 and lines 40-48 in col. 14).

With respect to claim 3, Dinker teaches the distributed store is configured to request the process to release the locked access, wherein the process is configured to release the locked access in response to said request (i.e., MSG_Reclaim_WrToken/RdToken message, lines 30-35 in col. 13 and lines 40-48 in col. 14).

With respect to claim 4, Dinker teaches the process is configured to release the locked access when the process no longer requires the locked access to the primary state (i.e., the client releases the access rights when the access rights are no longer required, lines 9-19 in col. 12).

With respect to claim 6, Dinker teaches the distributed store is configured to grant the locked access to the process executing in one of the application servers in response to a request for locked access from the process (lines 24-48 in col. 9 and lines 10-43 in col. 6).

With respect to claim 7, Dinker teaches while the process holds the locked access, the distributed store is configured to buffer one or more requests for locked access from one or more other processes executing within one or more of the plurality of application servers (i.e., a request queue, lines 15-24 and lines 52-56 in col. 14).

With respect to claim 8, Dinker teaches if the process releases the locked access to the primary state, the distributed store is configured to provide locked access to one of the other processes in response to the other process's buffered request (lines 15-24 and lines 52-56 in col. 14 and lines 18-35 in col. 11).

With respect to claim 9, Dinker teaches another process executing within one of the plurality of application servers is configured to request locked access to the primary sate from the distributed store, and wherein if no process currently hold locked access to the primary state, the distributed store is configured to provide locked access to the primary state to the other process (lines 24-48 in col. 9, lines 10-42 in col. 6, and lines 18-35 in col. 11).

The limitations of claims 17, 20, and 31 are rejected in the analysis of claim 1 above, and these claims are rejected on that basis.

The limitations of claims 19 and 23 are rejected in the analysis of claim 4 above, and these claims are rejected on that basis.

The limitations of claim 22 and 32 are rejected in the analysis of claim 3 above, and these claims are rejected on that basis.

The limitations of claims 24 and 33 are rejected in the analysis of claim 7 above, and these claims are rejected on that basis.

The limitations of claim 25 and 34 are rejected in the analysis of claim 8 above, and these claims are rejected on that basis.

Claim Rejections - 35 USC § 103

16. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

17. Claims 1, 6, 9, 17, 20, and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Montero et al. (U.S. Publication No. 2002/0143958) in view of Goldick (U.S. Publication No. 2003/0093457).

With respect to claim 1, Montero discloses a plurality of application servers, wherein each of the plurality of application servers is configured to access session data, wherein the session data represents the state of a client session for a client (fig. 1, abstract, section 11 on page 1, section 26 on pages 2-3, and section 36 on page 3). Montero discloses a common session database (a distributed store) comprising a primary state of the session data configured for access by the plurality of application servers and writes to the database controlled by a processing thread (fig. 1, section 35 on page 3, and section 40 on page 4). Montero does not explicitly disclose a locking management of the database. However, Goldick discloses a locking management on a

resource of a server system in a shared resource distributed computing environment (fig. 1, fig. 3, and sections 24-25 on page 3). Goldick discloses the server system comprising a resource configured for access by the plurality of client nodes, wherein the server system is configured to provide locked access to the resource to one of the plurality of the client nodes, wherein, while the resource is locked for the node, other nodes cannot access the resource (sections 24-25 on page 3). Goldick discloses wherein in providing locked access to the resource to the one of the plurality of the client nodes, the server system is configured to send a lock token to the node, wherein only the node that have received a lock token can access the resource (sections 44-45 on page 5 and fig. 3) in order to prevent data inconsistency (section 5 on page 1). Therefore, based on Montero in view of Goldick, it would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize teaching of Goldick to the system of Montero in order to prevent data inconsistency.

With respect to claim 6, Goldick discloses the server system is configured to grant the locked access to the one of the client nodes in response to a request for locked access from the node (sections 44-45 on page 5 and fig. 3). Therefore, the limitations of claim 6 are rejected in the analysis of claim 1 above, and the claim is rejected on that basis.

With respect to claim 9, Goldick teaches another node of the plurality of client nodes is configured to request locked access to the resource from the server system, and wherein if no node currently holds locked access to the resource, the server system is configured to provide locked access to the resource to the other node (section 44-45

on page 5 and fig. 3). Therefore, the limitations of claim 9 are rejected in the analysis of claim 1 above, and the claim is rejected on that basis.

The limitations of claims 17, 20, and 31 are rejected in the analysis of claim 1 above, and these claims are rejected on that basis.

18. Claims 2-3, 22, and 32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Montero et al. (U.S. Publication No. 2002/0143958) in view of Goldick (U.S. Publication No. 2003/0093457), and further in view of Eshel et al. (U.S. Publication No. 2003/0018785).

With respect to claim 2, Montero and Goldick do not explicitly disclose the process configured to hold locked access until after receiving a request to release the locked access. However, Eshel discloses after the node has completed a current access of a resource, the node is configured to hold locked access until after receiving a request to release the locked access (sections 11-12 on page 1 and fig. 2) in order to subsequently access the same resource without requesting additional locked access for the same resource. Therefore, based on Montero in view of Goldick, and further in view of Eshel, it would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teaching of Eshel to the system of Montero in order to subsequently access the same resource without requesting additional locked access for the same resource.

With respect to claim 3, Goldick further discloses the server system notifies a node holding a locked access that a break (of the locked access) is about to occur

(section 63 on page 7). Montero and Goldick do not explicitly disclose the distributed store configured to request the process to release the locked access. However, Eshel discloses a lock manager configured to request a node holding a locked access to release the locked access, wherein the node is configured to release the locked access in response to the request (sections 11-12 on page 1 and fig. 2) in order to provide the locked access to another node. Therefore, based on Montero in view of Goldick, and further in view of Eshel, it would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teaching of Eshel to the system of Montero in order to provide the locked access to another node.

The limitations of claims 22 and 32 are rejected in the analysis of claim 3 above, and these claims are rejected on that basis.

19. Claims 4, 7-8, 19, 23-25, and 33-34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Montero et al. (U.S. Publication No. 2002/0143958) in view of Goldick (U.S. Publication No. 2003/0093457), and further in view of Bennett (U.S. Patent No. 5,734,909).

With respect to claim 4, Montero and Goldick disclose the claimed subject matter as discussed above except releasing the locked access when the process no longer requires the locked access. However, Bennett teaches the process is configured to release the locked access when the process no longer requires the locked access to the resource (the primary state, lines 58-65 in col. 1, lines 7-16 in col. 2, line 54 in col. 3 thru line 14 in col. 4, lines 22-46 in col. 7, and lines 14-35 in col. 8) in order to allow another

process to access the resource. Therefore, based on Montero in view of Goldick, and further in view of Bennett, it would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teaching of Bennett to the system of Montero in order to allow another process to access the resource.

With respect to claim 7, Montero and Goldick disclose the claimed subject matter as discussed above except buffering requests. However, Bennett discloses while the process holds the locked access, the central server (the distributed store) is configured to buffer one or more requests for locked access from one or more other processes executing within one or more of the plurality of client nodes (application servers, lines 58-65 in col. 1 line 54 in col. 3 thru line 14 in col. 4, lines 7-46 in col. 7, lines 53-67 in col. 7, and lines 14-35 in col. 8) in order to award processes with a locked request in the sequence that lock requests arrive. Therefore, based on Montero in view of Goldick, and further in view of Bennett, it would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teaching of Bennett to the system of Montero in order to award processes with a locked request in the sequence that lock requests arrive.

With respect to claim 8, Bennett further teaches if the process release the locked access to the resource (the primary state), the central server (the distributed store) is configured to provide locked access to one of the other processes in response to the other process's buffered request (lines 58-65 in col. 1 line 54 in col. 3 thru line 14 in col. 4, lines 7-46 in col. 7, lines 53-67 in col. 7, and lines 14-35 in col. 8). Therefore, the

limitations of claim 8 are rejected in the analysis of claim 7 above, and the claim is rejected on that basis.

The limitations of claims 19 and 23 are rejected in the analysis of claim 4 above, and these claims are rejected on that basis.

The limitations of claims 24 and 33 are rejected in the analysis of claim 7 above, and these claims are rejected on that basis.

The limitations of claims 25 and 34 are rejected in the analysis of claim 8 above, and these claims are rejected on that basis.

20. Claims 5, 18, and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Montero et al. (U.S. Publication No. 2002/0143958) in view of Goldick (U.S. Publication No. 2003/0093457), and further in view of Bender et al. (US 2003/0163494 A1).

With respect to claim 5, Montero and Goldick disclose the claimed subject matter as discussed above except providing locked access to a thread within a process. However, Bender discloses the process is configured to provide locked access to at least a portion of a resource to a thread executing within the process, wherein, while the at least a portion of the resource is locked for the thread, other threads executing within the process cannot access the at least a portion of the resource (abstract, section 12 on page 12, sections 33-34 on page 3, sections 37-41 on page 4, and sections 43-45 on page 5). Therefore, based on Montero in view of Goldick, and further in view of Bender, it would have been obvious to one having ordinary skill in the art at the time the

invention was made to utilize the thread-level locking mechanism of Bender to the system of Montero in order to avoid data inconsistency.

The limitations of claims 18 and 21 are rejected in the analysis of claim 5 above, and these claims are rejected on that basis.

21. Claims 26, 28-30, 35, and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Montero et al. (U.S. Publication No. 2002/0143958) in view of Bennett (U.S. Patent No. 5,734,909) and Bender et al. (U.S. Publication No. 2003/0163494 A1), and further in view of Eshel et al. (U.S. Publication No. 2003/0018785).

With respect to claim 26, Montero discloses a process executing within one of a plurality of application servers, wherein each of the plurality of application servers is configured to access session data, wherein the session data represents the state of a client session for a client (fig. 1, abstract, section 11 on page 1, section 26 on pages 2-3, and section 36 on page 3). Montero discloses a common session database (a distributed store) comprising a primary state of the session data configured for access by the plurality of application servers and writes to the database controlled by a processing thread (fig. 1, section 35 on page 3, and section 40 on page 4). Montero does not explicitly disclose a locking management of the database. However, Bennett discloses a locking management on a resource of a central server in a shared resource distributed computing environment, wherein the resource of the central server is updated or synchronized with data from clients (abstract, line 15 in col. 1 thru line 30 in

col. 2, and line 8 in col. 3 thru line 60 in col. 4). Bennett discloses the resource configured for access by the plurality of client nodes, wherein a lock is granted to a process requesting locked access and executing within one of the plurality of the client nodes, wherein, while the resource is locked for the process, other processes cannot access the resource (line 8 in col. 3 thru line 60 in col. 4, and lines 4-20 in col. 7). Therefore, based on Montero in view of Bennett, it would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize a resource locking mechanism of Bennett to the system of Montero in order to avoid data inconsistencies. Montero and Bennett do not explicitly disclose providing locked access to a thread within a process. However, Bender discloses the process holding the lock granting a thread-level lock to a portion of a resource to a thread running within the process, wherein, while the thread holds the thread-level lock, other threads cannot access the portion of the resource (abstract, section 12 on page 12, sections 33-34 on page 3, sections 37-41 on page 4, and sections 43-45 on page 5). Therefore, based on Montero in view of Bennett, and further in view of Bender, it would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the thread-level locking mechanism of Bender to the system of Montero in order to avoid data inconsistencies. Montero, Bennett, and Bender do not explicitly disclose requesting the process to release the locked access. However, Eshel discloses requesting a node holding a locked access to release the locked access, wherein the node is configured to release the locked access in response to the request (sections 11-12 on page 1 and fig. 2) in order to provide the locked access to another node.

Therefore, based on Montero in view of Bennett and Bender, and further in view of Eshel, it would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teaching of Eshel to the system of Montero in order to provide the locked access to another node.

With respect to claim 28, Bennett teaches the process is configured to release the locked access when the process no longer requires the locked access to the resource (the primary state, lines 58-65 in col. 1 line 54 in col. 3 thru line 14 in col. 4, lines 22-46 in col. 7, and lines 14-35 in col. 8). The limitations are rejected in the analysis of claim 26 above, and the claim is rejected on that basis.

With respect to claim 29, Bennett discloses while the process holds the locked access, buffering one or more requests for locked access from one or more other processes executing within one or more of the plurality of client nodes (application servers, lines 58-65 in col. 1 line 54 in col. 3 thru line 14 in col. 4, lines 7-46 in col. 7, lines 53-67 in col. 7, and lines 14-35 in col. 8). The limitations are rejected in the analysis of claim 26 above, and the claim is rejected on that basis.

With respect to claim 30, Bennett teaches if the process release the locked access to the resource (the primary state), the central server (the distributed store) is configured to provide locked access to one of the other processes in response to the other process's buffered request (lines 58-65 in col. 1 line 54 in col. 3 thru line 14 in col. 4, lines 7-46 in col. 7, lines 53-67 in col. 7, and lines 14-35 in col. 8). The limitations are rejected in the analysis of claim 29 above, and the claim is rejected on that basis.

The limitations of claim 35 are rejected in the analysis of claim 26 above, and the claim is rejected on that basis.

With respect to claim 37, Bennett teaches the process is configured to release the locked access when the process no longer requires the locked access to the resource (the primary state, lines 58-65 in col. 1, line 54 in col. 3 thru line 14 in col. 4, lines 22-46 in col. 7, and lines 14-35 in col. 8). The limitations are rejected in the analysis of claim 35 above, and the claim is rejected on that basis.

Allowable Subject Matter

22. Claim 10 would be allowable if rewritten or amended to overcome the claim objections and a terminal disclaimer is filed, set forth in this Office action.

23. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Joon H. Hwang whose telephone number is 571-272-4036. The examiner can normally be reached on 9:30-6:00(M~F).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, HOSAIN T. ALAM can be reached on 571-272-39783978. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Joon Hwang
Patent Examiner
Technology Center 2100

11/30/06